

半结构化数据模型与面向 Internet 的数据挖掘技术

Semi-Structure Data Model and the Technique of Data Mining for Internet-Oriented

陈一明

(茂名学院师范学院 广东茂名525000)

Abstract In order to go on the data mining from the Internet's large number of data, it must be based on semi-structure model of data and the technique for its extraction. In this article, the author analyses the technique of data mining and the characteristic of data's structure on Internet. At the same time, since the XML is regarded as a data model of semi-structure, which is queried and extracted accurately, thereby the data mining of Internet-oriented is finished, he gives discussion to the method connecting with the application on SQL Server 2000.

Keywords XML, Data mining, Internet, Data model

随着计算机技术的飞速发展,人们利用信息技术生产和搜集数据的能力在逐步提高,数据库技术被应用于社会各行各业。但是,数据量的不断增加,我们越来越有一种被数据淹没的感觉。如何利用我们所面对的(特别是 Internet 上的)大量数据,从中发现有用的知识,使它们为企业的业务决策和战略发展服务是业界共同的课题。

1 数据挖掘与半结构化数据模型

1.1 数据挖掘与异构数据库

数据挖掘(DM)是从大型数据库或数据仓库中发现并提取隐藏在其中的模式或者知识的过程。其目的是帮助分析人员寻找数据之间的关联,发现被忽略的要素,而这些信息对于预测趋势和决策行为是十分有用的。DM 的一般过程如图1。

数据挖掘源于人工智能的机器学习领域,其实质的内涵是在一个已知状态的数据集(Data Set)上,通过设定一定的学习算法,从数据集中获取所谓的知识。知识发现技术的相关研究为数据挖掘技术提供了坚实的理论基础,数据挖掘技术实际上是人工智能领域中的知识发现技术和数据库技术共同发展的结果。

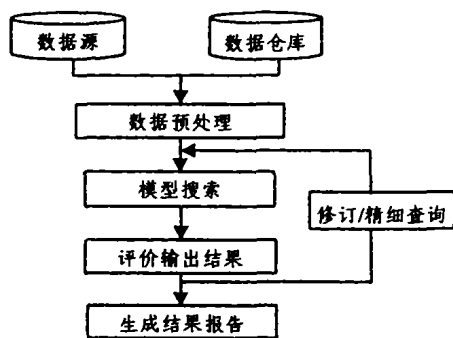


图1 数据挖掘的一般过程

传统意义的数据库、数据仓库和 Internet 中的数据是我们所拥有的数据资源,也是进行数据挖掘的基础。传统数据库中的数据结构性很强,其中的数据为完全结构化的数据;数据仓库是由数据库导出的用于分析型的数据环境;我们可将 Web 网站上的信息看作一个数据库,一个更大、更复杂的数

据库,Web 上的每一个站点就是一个数据源,每个数据源都是异构的,因而每一站点之间的信息和组织都不一样,这就构成了一个巨大的异构数据库环境。

1.2 半结构化数据模型与数据挖掘

Internet 上的数据与传统的数据库中的数据不同,传统的数据库都有一定的数据模型,可以根据模型来具体描述特定的数据,同时可以很好地定义和解释相关的查询语言。而 Internet 上的数据非常复杂,没有特定的模型描述,每一站点的数据库都各自独立设计,并且数据本身具有自述性和动态可变性,其结构也不可琢磨,是一种我们称之为半结构化的数据。所谓半结构化是相对于结构化(传统数据库)和非结构化(如一本书、一张图片等)而言的。但是 Internet 上存在的数据既不是完全结构化的也不是完全非结构化的,因为它的页面也具有一定的描述层次,存在一定的结构,所以我们将它称为半结构化的数据。

如果想要利用 Internet 上的数据进行数据挖掘,必须先要研究站点之间异构数据的集成问题,只有将这些站点的数据库都集成起来,提供给用户一个统一的视图,才有可能从巨大的数据资源中获取所需的東西。其次,还要解决 Web 上的数据查询问题,因为如果所需的数据不能很有效地得到,对这些数据进行分析、集成、处理就无从谈起。针对 Internet 上的数据半结构化的特点,寻找一个半结构化的数据模型则成为了解决上述问题的关键所在。此外,除了要定义这样一个半结构化数据模型外,还需要一项技术能够自动地从现有数据中将这个模型抽取出来,这就是所谓的模型抽取技术。因此,半结构化数据模型及其抽取技术是面向 Internet 的数据挖掘技术实施的前提。

2 XML 与面向 Internet 的数据挖掘

XML 是可扩展的标记语言,它的最大特点在于其标志(TAG)不是预先定义的,是由用户自己定义,能够反映一定的数据含义。简单来讲,XML 是一种半结构化的数据模型,它解决了 HTML 不能解决的两个 Web 问题,即 Internet 发展速度快而接入速度慢的问题,以及可利用的信息多,但难以找到自己需要的那部分信息的问题。XML 具有简单、开放、高效且可扩充和国际化等特点,这些特点决定了其卓越的性能表

陈一明 讲师,从事计算机教学与应用研究。

现。因此,XML 将成为在 Internet 上的数据查询和模式抽取的一个重要途径。

XML 已经成为正式的规范,可以用 XML 的格式标记和交换数据。使用 XML 可升级的三层架构模型可以从存在的数据中产生出来,使用 XML 结构化的数据可以从商业规范和表现形式中分离出来。数据的集成、发送、处理和显示是下面过程中的每一个步骤(如图2)。

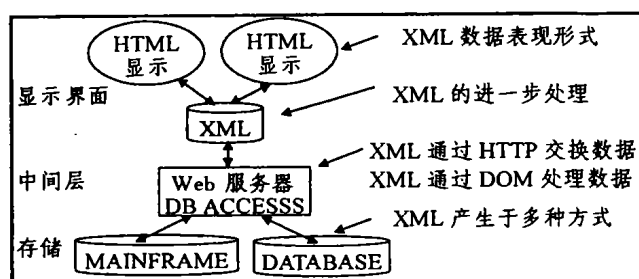


图2 XML 三层架构模型

XML 能够使不同来源的结构化的数据很容易地结合在一起。软件代理商可以在中间层的服务器上对从后端数据库和别处来的数据进行集成。然后,数据就能被发送到客户或其他服务器做进一步的集合、处理和分发。XML 的扩展性和灵活性允许它描述不同种类应用软件中的数据,从描述搜集的 Web 页到数据记录,从而可通过多种应用得到数据。同时,由于基于 XML 的数据是自我描述的,用户可以方便地进行本地计算和处理,XML 格式的数据发送给客户后,客户可以用应用软件解析数据并对数据进行编辑和处理。XML 文档对象模式(DOM)允许用脚本或其他编程语言处理数据,数据计算不需要回到服务器就能进行。XML 可以被用来分离使用者观看数据的界面,使用简单灵活开放的格式,可以给 Web 创建功能强大的应用软件。另外,数据发到桌面后,能够用多种方式显示。

XML 还可以通过以简单开放扩展的方式描述结构化的数据。由于数据显示与内容分开,XML 定义的数据允许指定不同的显示方式,使数据更合理地表现出来。本地的数据能够以客户配置、使用者选择或其他标准决定的方式动态地表现出来。通过 XML,数据可以粒状地更新,每当一部分数据变化后,不需要重发整个结构化的数据。变化的元素必须从服务器发送给客户,变化的数据不需要刷新整个使用者的界面就能够显示出来。XML 也允许加进其他数据,加入的信息能够进入存在的页面,不需要浏览器重新发一个新的页面。XML 是用户与不同数据源进行交互的标准语言,它的自定义性及可扩展性足以表达各种类型的数据。客户收到数据后可以进行处理,也可以在不同数据库间进行传递。总之,XML 解决了数据的统一接口问题。但是,与其他的数据传递标准不同的是,XML 并没有定义数据文件中数据出现的具体规范,而是在数据中附加 TAG 来表达数据的逻辑结构和含义。这使 XML 成为一种程序能自动理解的规范。

XML 应用于将大量运算负荷分布在客户端,即客户可根据自己的需求选择和制作不同的应用程序以处理数据,而服务器只须发出同一个 XML 文件。应用 XML 将处理数据的主动权交给了客户,改变传统的“Client/Server”由客户向服务器发出不同的请求,服务器分别予以响应的工作方式;服务器所作的只是尽可能完善、准确地将数据封装进 XML 文件中。XML 的自解释性使客户端在收到数据的同时也理解数据的

逻辑结构与含义,从而使广泛、通用的分布式计算成为可能。

XML 还被应用于网络代理,以便对所取得的信息进行编辑、增减以适应个人用户的需要。此外,XML 文件中还可以包含诸如难度系数、往年错误率等其他相关信息,这样只需几个小程序,同一个 XML 文件便可变成多个文件传送到不同的用户手中。

3 基于 SQL Server 2000 的 XML 在 Web 中的应用

支持 XML 的数据库能够在 XML 文档和数据库之间进行数据的传输。通常是设计成为能够存储和提取以数据为中心的文档,在把 XML 文档进行解析以后,存储到相应的表格中。当然,也可以存储以文档为中心的文档,就是把整个文档作为一个单一的表中的一个字段,然后通过文本检索机制进行查询。因为许多数据库现在能够把内容发布到网站上,基于 XML 的数据库和 XML 服务器之间的差别就变得很模糊。Microsoft SQL Server 2000 的新特性之一就是支持 XML。

3.1 使用 HTTP 访问 SQL Server 2000 数据库

可以通过 HTTP 方式直接访问具备 XML 功能的 SQL Server 2000 数据库。访问 SQL Server 2000 的 HTTP 允许进行如下活动:

1) 在 URL 中直接使用 SQL 语句。例如:

http://IISServer/nwind?sql=SELECT + " * + FORM + Customers + FOR + XML + AUTO & root=root

该语句中 FOR XML 子句表示返回的结果是一个 XML 文档,而不是标准的结果集。Root 参数表明了单个的顶层元素。

2) 直接在 URL 中指定模板。模板包含了一条或多条 SQL 语句的 XML 文档,它允许从一个有效的 XML 文档中把数据集中到一起获取,这样就不需要直接在 URL 中指定查询语句。

3) 在 URL 中指定模板文件。可以使用存储在文件中的模板,在 URL 中直接指定一个模板文件,避免在 URL 中写一段很长的 SQL 查询语句,并且还可以加强系统的安全。

4) 在简化 XML 数据计划上使用 Xpath 查询。在映射计划上编写 Xpath 查询在概念上就如同使用 CREATE VIEW 语句创建了视图,然后编写 SQL 语句来查询它们一样。

5) 在 URL 中直接指定数据库对象。数据库对象(如表和视图)可以指定为 URL 的一部分,Xpath 也可以在数据库对象上指定。

3.2 读取 XML 数据

在 SQL Server 2000 中可以使用执行 SQL 语句返回 XML 数据,而不是标准的数据行。这样的查询可以直接执行,也可以通过存储过程执行。要直接获取结果集可以在 SELECT 语句中使用 FOR XML 子句,可以为该子句指定三种 XML 的模式:RAW、AUTO 或 EXPLICIT。例如:下面的 SELECT 语句获取了 Northwind 数据库中 Customers 表和 Orders 表中的信息,查询中为 FOR XML 子句指定了 AUTO 模式:

```
SELECT Customers.CustomerID,ContactName,Company Name,
       Orders.CustomerID,OrderDate
FROM Customers,Orders
WHERE Customers.CustomerID = Orders.CustomerID
      AND(Customers.CustomerID=N'ALFKI'
          OR Customers.CustomerID = N'XYZAA')
ORDER BY Customers.CustomerID
FOR XML AUTO
```

(下转第103页)

请求的 Agent 并发送一个点到点消息进行应答,这个消息包含了查找服务的代理对象,Agent 可通过它来访问查找服务。具体实现如图2。



图2 Agent 加入联盟及与其它 Agent 的交互

提供服务的 Agent2 通过发现协议找到查找服务(Lookup Service),再由加入协议将所提供的服务注册到查找服务中,要注册的信息包括服务标识号、服务实体(包含服务类型的信息)和服务属性。使用服务的 Agent1 也通过发现协议找到查找服务,再由查找协议按所需服务的类型和属性查询服务,然后 Agent1 以特定的服务协议与 Agent2 通信以使用 Agent2 提供的服务。

对于查找服务,没必要与各 Agent 联盟一对一地映像,网络上的每个查找服务都可以为一个或多个 Agent 联盟提供服务,每个 Agent 联盟也可以有一个或多个查找服务支持。这样就有效地消除了单点故障(Single Point Failure)。

基于 Jini 的多 Agent 系统中,Agent 模型如图3所示。

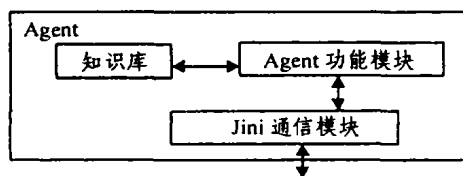


图3 基于 Jini 的 Agent 模型

Agent 的知识库包括 Agent 对自身的描述、Agent 对世界状态的描述及对其它 Agent 状态的描述。功能模块主要是根据 Agent 自身的目标完成规划、调度和动作,功能模块是 Agent 的核心,因为它反映了 Agent 的真正功能和决定了 Agent 的结构,根据 Agent 功能模块的不同,Agent 可为慎思型 Agent、反应型 Agent 和混合型 Agent^[6],以适应不同的需要。Jini 通信模块用来与其它 Agent 进行通信,通信的内容用 Agent 通信语言(ACL)来编码,如可以使用 Agent 通信语言事实上的标准语言 KQML 或 FIPA ACL。

4 分析

Jini 提供了可用来实现动态 Agent 联盟的机制。通过使

用租借,一个 Agent 可以表明它加入一个 Agent 联盟的时间段。在租借结束时,Agent 可以考虑重新租借或退出 Agent 联盟。这样一个 Agent 可动态地加入和退出 Agent 联盟,可以很容易地实现动态、开放的多 Agent 系统。Jini 技术的另一个对实现多 Agent 系统有用的特性是一个要加入 Agent 联盟的 Agent 不一定要必须运行在 Java 虚拟机(JVM)上,这个 Agent 可以使用一个代理(proxy)来作为与其它 Agent 及查找服务交互的中介。通过代理的使用,一个 Agent 可以透明地和另一个 Agent 进行通信;只需预先知道另一个 Agent 有什么功能,而不必了解该 Agent 是如何去做的,包括实现该 Agent 的语言、软硬件平台、物理位置和使用的通信协议。这样 Agent 就可以不用 Java 实现,甚至能用代理来封装遗产软件(legacy software),实现异构的多 Agent 系统。

另外,Jini 还提供了可以用来方便地实现基于黑板通信的多 Agent 系统的服务—JavaSpaces。JavaSpaces 是建立在基本 Jini 框架之上的特殊服务,它为 Java 对象提供存储功能。利用 JavaSpaces, Jini 服务和客户可以创建并控制 Java 对象的“空间”,它们可以为 Java 对象得到租借的存储区,可以在存储空间内搜寻存储的对象或删除存储对象。这样一组 Agent 可以使用 JavaSpaces 作为“共享黑板”来存储和取出对象。对“空间”的所有修改操作都以事务安全的方式进行,多个 Agent 可以使用 Jini 的事务机制来实现多个操作或多个“空间”更新的原子性。在空间中存放的对象可以表示要做的工作(因此可用于创建生产者/消费者模式的多 Agent 系统),或只表示数据信息(可用于创建有分布式投票功能或报价的多 Agent 系统)。

结论 理想的多 Agent 系统构造平台对于实现开放、动态、异构的多 Agent 系统的重要性是不言而喻的。Jini 提供了构建简单、灵活、开放的分布式系统的基础,而 Agent 技术是构建复杂、“智能”软件最有潜力的技术,基于 Jini 的多 Agent 系统框架能够结合 Agent 技术和 Jini 技术的优点,使构建面向 Internet 的开放、复杂的软件变得更为高效。

参考文献

- Jennings N R, Wooldridge M. Agent-Oriented Software Engineering. in Handbook of Agent Technology (ed. J. Bradshaw) AAAI/MIT Press, 2000(to appear)
- Sycara K. MultiAgent Systems. AI Magazine, 1998, 19(2): 79~92
- Weiss G. MultiAgent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, 1999
- Edwards W K, Jini C. Prentice Hall PTR 1999
- Technology J. Available at <http://www.sum.com/Jini>
- Wooldridge M, Jennings N R. Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 1995, 10(2)

(上接第91页)

3.3 编写对 XML 文档的查询语句

Transact-SQL 关键字 OPENXML 为内存中的 XML 文档提供了数据行。OPENXML 与表或视图是比较相似的,提供了对 XML 数据的访问途径。在数据行提供者(表、视图或 OPENROWSET)可以作为数据源出现的地方,都可以在 SELECT 或 SELECT INTO 语句中使用 OPENXML 关键字。要使用 OPENXML 编写对文档的查询语句,必须首先调用 sqxml-preparedocument 系统过程分解 XML 文档,并得到对分解后文档的处理。使用系统存储过程可以释放内存中的 XML 文档。

结束语 面向 Internet 的数据挖掘是一项复杂的技术,而 XML 是直接面对 Web 数据的,不仅可以很好地兼容原有

的 Web 应用,而且可以更好地实现 Web 中的信息共享与交换。XML 可看作一种半结构化的数据模型,可以很容易地将 XML 的文档描述与关系数据库中的属性对应起来,实施精确的查询与模型抽取。微软的 SQL Server 2000 支持完全集成的 XML 环境,使面向 Internet 的数据挖掘成为可能。

参考文献

- 罗运模. SQL Server 2000 数据仓库应用与开发. 北京:人民邮电出版社, 2001.7
- 长城工作室数据组. SQL Server 2000 高级应用. 北京:人民邮电出版社, 2001.8
- Inmon, W. H. 著,王志海等译. 数据仓库. 北京:机械工业出版社, 2000.5
- Lou Agosta 著,潇湘工作室译. 数据仓库技术指南. 北京:人民邮电出版社, 2000.11